

Creación de una Ontología para el mundo de la restauración

Roman Boix Rodríguez

Resumen—Las ontologías se definen como especificaciones explícitas de conceptos en un determinado dominio y las relaciones entre éstos, además permiten hacer suposiciones y extraer nuevo conocimiento a partir de lo que se conoce. Es por ello que este proyecto tiene como objetivo desarrollar una primera versión de una ontología para el mundo de la restauración. Ésta se utilizará para dar soporte a un conjunto de aplicaciones, dentro de la empresa Delectatech, destinadas a la gestión de negocios dentro del mundo de la restauración.

El proyecto se ha dividido en diferentes partes consecutivas, la obtención de una base de conocimiento, el desarrollo de una herramienta para validar esta base de conocimiento por parte de expertos, la creación de la ontología, la elaboración de axiomas y finalmente la creación de demostraciones.

Palabras clave—Ontología, axioma, inferencia, restauración, alimentos, recetas, ingredientes, protégé, procesamiento de lenguaje natural, ontology web language, sparql, semantic web rule language.

Abstract—Ontologies are defined as explicit specifications of particular domain concepts and relationships between these, they also allow making assumptions and extract new knowledge from what is known. That is why this project aims to develop a first version of an ontology for the culinary field. This will be used to support a set of applications, owned by Delectatech a business focused on the management within the restoration world.

The project has been divided into different consecutive parts, such as obtaining a knowledge base, the development of a tool to validate this knowledge base by experts, the creation of an ontology, the development of axioms and finally the creation of demonstrations.

Index Terms— Ontology, axiom, culinary, food, recipes, ingredients, protégé, natural language processing, ontology web language, sparql, semantic web rule language.



1 INTRODUCCIÓN

Teniendo en cuenta la importancia del sector de la hostelería en la economía global, es curioso ver como la restauración es uno de los sectores menos productivos de toda Europa y donde menos se ha innovado con las TIC. De acuerdo a las escuelas de restauración y a los profesionales del sector, la razón de esta falta de productividad pasa por:

- Poco control de la rentabilidad de la oferta gastronómica y del control de costes. A los restaurantes les cuesta mucho saber cuánto ganan por cada plato y, sobretudo, cuánto les cuesta producirlo.
- Falta de información segmentada sobre el mercado y la competencia adaptada a cada negocio. En un sector donde los precios cambian día a día y las opiniones de los comensales pueden cerrar un restaurante, es vital tener información del exterior para poder tomar las decisiones correctas en todo momento.
- Una falta de control de un producto perecedero, muy complicado de gestionar y que además provocan desperdicios de miles de toneladas de comida al año.
- Falta de formación y profesionalización. Una gran mayo-

ría de hosteleros no tienen la formación adecuada ni experiencia en el sector.

Es por ello que nace la necesidad de crear un motor de inteligencia artificial capaz de extraer conocimiento (a partir de comentarios, blogs, recetas, etc.) y aprender de forma autónoma todo tipo de información relacionada con la restauración y la gastronomía. De este modo, ser capaz de inferir todo tipo de conceptos acerca del mundo de la restauración, como por ejemplo la mejor combinación de alimentos, que ingrediente puede substituir a otro o que alérgenos tiene una determinada receta.

1.1 Contexto

Es importante destacar que el trabajo íntegro se ha desarrollado en la empresa Delectatech, la cual pretende crear una plataforma para el sector de la restauración con la intención de ayudar a los negocios de hostelería a mejorar su productividad y a conocer el impacto que producen sus decisiones en sus clientes.

2 OBJETIVOS

Como se ha comentado anteriormente el objetivo de este TFG era construir un sistema inteligente capaz de aprender y hacer sugerencias, es decir, la creación de una ontología alimenticia.

Para alcanzar este objetivo se definieron diferentes sub—

objetivos:

- Obtener la base de conocimiento: uno de los puntos más importantes para la creaci3n de la ontología es definir correctamente su taxonomía.
- Crear la primera versi3n de la ontología: directamente relacionado con el objetivo global del TFG es la creaci3n de una primera versi3n de la ontología.
- Validar la base de conocimiento: Este sub-objetivo es importante ya que es necesario que la taxonomía de la ontología permita clasificar cualquier ingrediente.
- Elaborar axiomas y hacer inferencias: Es importante dejar algunos ejemplos y axiomas ya hechos para futuros desarrollos con la ontología.
- Implementar demostraciones: Para comprobar el funcionamiento y la utilidad de los diferentes axiomas.

En la Tabla 1 se muestran los diferentes sub-objetivos.

Objetivo	Tipo
Obtener la base de conocimiento	CRITICO
Validar la base de conocimiento	PRIORITARIO
Crear la primera versi3n de la ontología	CRITICO
Elaborar axiomas y hacer inferencias	SECUNDARIO
Implementar Demostraciones	SECUNDARIO

Tabla 1 Objetivos del Proyecto

3 METODOLOGÍA

A continuaci3n, se explicará la metodología de trabajo seguida y la planificaci3n.

3.1 SCRUM

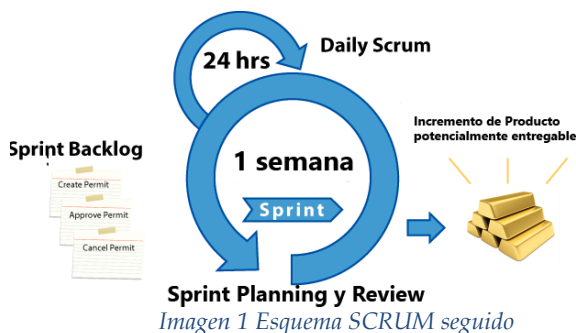
Debido a que el desarrollo de este trabajo se realizó dentro de la empresa Delectatech, se consideró que lo más conveniente era adaptarse a su metodología de trabajo. Concretamente, la metodología usada ha sido SCRUM.

Los diferentes parámetros de la metodología escogida fueron los siguientes:

- Sprint: se definieron con una duraci3n de una semana.
- Sprint planning: se han realizado al principio de cada Sprint. Se definían los objetivos y tareas (Sprint Backlog) a cumplir durante la semana.
- Sprint review: Realizados al principio de cada Sprint, se revisaba el estado final del pasado Sprint.
- Daily Scrum: no se realizan formalmente debido al tamaño del equipo de desarrollo, en cambio al principio o al finalizar el día se comenta con el Scrum master el estado de las tareas y la carga de trabajo.

Partiendo de la estructura de la empresa, los roles estarán definidos de la siguiente manera:

- Scrum master: responsable técnico dentro de la empresa
- Product Owner: fundadores de la empresa.
- Dev-Team: el equipo completo dentro de la empresa.



3.2 Planificaci3n

Debido a la utilizaci3n de la metodología ágil Scrum, no existía una planificaci3n estricta, ya que las tareas a desarrollar iban cambiando según las necesidades del proyecto en cada momento.

Así que se definieron unos hitos y una lista de tareas que fue cambiando a medida que el proyecto avanzaba.

Los hitos definidos y la lista de tareas definitivas se pueden consultar en A1 Log planificaci3n y A2 Hitos

4. BACKGROUND

A continuaci3n, se exponen los conceptos mínimos imprescindibles del mundo de las ontologías: que son, que tipos hay, diferentes metodologías para implementarlas y lenguajes y diferentes maneras de realizar axiomas.

4.1 Ontología

4.1.1 Que es una Ontología

Una de las definiciones que más sentido tiene para este proyecto viene dada por Thommas R. Gruber [1]:

"An ontology is an explicit specification of a conceptualization (an abstract, simplified view of the world that we wish to represent for some purpose)."

4.1.2 Tipos de Ontologías

Las ontologías se pueden clasificar de múltiples maneras [2] dependiendo del grado de generalidad. Para este trabajo, la ontología que se va a implementar se podría clasificar como de dominio, ya que captura el conocimiento de un dominio en particular, concretamente, se centra en el dominio de la restauraci3n y alimentaci3n.

4.1.3 Metodologías para crear Ontologías

Existen diferentes metodologías para crear Ontologías [3] [4] [5] [6] [7] [8] [9] [10] [11]. De todas ellas, la que mejor se adaptaba a lo que se quería hacer es la metodología CYC, la cual define algunos pasos generales para la construcci3n de ontologías, concretamente:

1. Extracci3n manual del conocimiento común que está implícito en diferentes fuentes.
2. Una vez extraído suficiente conocimiento, se adquiere nuevo conocimiento usando herramientas

de procesamiento de lenguaje natural o aprendizaje computacional.

4.2 Ontology Web Language 2

4.2.1 Que es OWL 2

Es un lenguaje basado en lógica computacional, de forma que el conocimiento expresado en OWL se puede procesar y razonar mediante Software, ya sea para verificar la consistencia de ese conocimiento o para hacer explícito el conocimiento implícito. Concretamente es el lenguaje usado para la creación de la ontología de este trabajo. [12] [13] [14]

4.2.2 Nociones básicas para el modelado de conocimiento

Para representar el conocimiento utilizamos:

- Axiomas: los enunciados básicos que se pueden expresar
- Entidades: elementos usados para representar objetos del mundo real
- Expresiones: combinación de entidades, de forma que se crean descripciones más complejas a partir de las básicas.

4.2.3 Open World Assumption

Es el supuesto de que el valor de un axioma puede ser cierto independientemente de si es o no es conocido para ser cierto. Declaraciones sobre conocimiento que no está incluido o inferido a partir del conocimiento explícito del sistema, puede considerarse desconocido, en lugar de falso o erróneo.

Dentro de OWL, la ausencia de un axioma en particular significa que un razonador no puede (y no debe) inferir que la afirmación es falsa.

4.3 Razonadores

Principalmente, proporcionan un conjunto de operaciones a partir de los axiomas definidos en la ontología, es decir, son los encargados de extraer automáticamente las consecuencias de los diferentes axiomas presentes. [15]

4.4 Semantic Web Rule Language

Es un lenguaje de reglas basado en OWL que permite escribir reglas para conseguir un razonamiento deductivo mucho mayor. Semánticamente está construido sobre la misma base lógica de descripción que OWL. [16]

4.5 Protégé

Es un editor y gestor de conocimiento, gratuito y de código libre. Proporciona una interfaz gráfica para definir ontologías además de incluir diferentes razonadores para validar que los diferentes modelos cargados sean consistentes e inferir nuevo conocimiento.

Desarrollado por la Universidad de Standford y la Universidad de Manchester.

5 ESTADO DEL ARTE

Previamente al desarrollo del proyecto se buscaron otras herramientas y aplicaciones que también usaran ontologías relacionadas con la alimentación i/o restauración. Las dos aplicaciones que más se parecen a lo que nosotros

nos proponíamos hacer eran Chef Watson [17] y Spoonacular¹.

5.1 Chef Watson

Chef Watson es una herramienta implementada por IBM con la colaboración de la web especializada *Bon Appétit*². Esta herramienta está basada en Watson de IBM, un sistema capaz de responder preguntas expresadas en lenguaje natural.

La parte que más nos interesa es la parte correspondiente a las fuentes de información, ya que Watson utiliza muchísimas fuentes de información, entre ellas diferentes ontologías como DBpedia, WordNet y YAGO.

Para Chef Watson además de las anteriores se ha usado toda la información de *Bon Appétit*, concretamente más de 10.000 recetas con sus correspondientes ingredientes.

La funcionalidad principal de Chef Watson es la de sugerir nuevas recetas a partir de combinaciones de sabores. Para ello dice analizar la sinergia de los compuestos que forman el sabor de cada una de las recetas y combinaciones de ingredientes.

A favor

Como punto principal es la cantidad de información que contiene y la clasificación de ésta.

En contra

A primera vista el sistema parece cerrado, es decir, no parece proporcionar otros servicios que el que muestra. No parece ser un sistema susceptible a cambios o a añadir nuevas funcionalidades. Por otro lado, no ofrece ninguna manera de acceder ni a la información ni a ninguna de sus funcionalidades a través de ninguna API ni ningún sistema parecido.

Por último, no parece que IBM haya desarrollado la herramienta para nada más que para hacer publicidad de la capacidad que tiene Watson.

5.2 Spoonacular

Según la propia web de Spoonacular se trata de un sistema de gestión de alimentos.

Actualmente ofrece un planificador de comidas, a partir del cual se puede generar la lista de la compra y calcular la información nutricional.

A favor

La cantidad de ingredientes y recetas en este caso también es muy grande, aunque el punto más interesante en este caso, es la existencia de productos de marcas, como Coca-Cola o Nestlé y toda su información nutricional.

Otro punto que posiblemente sea muy beneficioso para nosotros en un futuro es el hecho de disponer de una API para acceder a todo su contenido.

En Contra

El principal problema de este sistema es que no ofrece una clasificación de los productos, recetas e ingredientes que dispone, por lo tanto, para el desarrollo de este trabajo tampoco ha sido de mucha utilidad.

¹ Spoonacular. <https://spoonacular.com/>

² Bon Appetit: <http://www.bonappetit.com/>

6 DESARROLLO DEL PROYECTO

El desarrollo del proyecto se dividió en dos partes muy diferenciadas. Por un lado, el análisis de diferentes fuentes de información para la creación de la base de conocimiento y de la herramienta FoodMine. Y, por otro lado, la transformación de la base de conocimiento en ontología, la elaboración de axiomas y de la implementación de la aplicación de creación de recetas.

Es importante destacar lo que se entiende por base de conocimiento. En el contexto de este trabajo, la base de conocimiento es una base de datos con la información que posteriormente se transformará en la ontología. La diferencia con la ontología es que la base de datos simplemente almacena datos, en cambio en la ontología, también permite hacer inferencias y extraer conocimiento nuevo.

6.1 Obtención de la base de conocimiento

El objetivo era construir una jerarquización de categorías de ingredientes lo suficientemente óptima para clasificar cualquier ingrediente que pudiera aparecer en cualquier receta. Además de conseguir un conjunto de ingredientes básicos con los que inicializar la ontología.

En un principio la búsqueda se basaba exclusivamente en fuentes online, concretamente en diferentes webs de recetas donde la lista de ingredientes era inmensa. El problema principal por el cual se desestimó este método fue la dificultad que provocaba el lenguaje natural. Cada receta la escribía una persona diferente, por lo tanto, para un mismo ingrediente se encontraron diferentes maneras de expresarlo, como, por ejemplo, en el caso de la mayoría de pescados como se muestra en la Tabla 2.

<i>Merluccius merluccius</i>	<i>Brama brama</i>	<i>Dicentrarchus labrax</i>
Merluza	Palometa	Lubina
Carioca	Castañeta	Robalo
Alambolo	Japuta	Lubaro
Pescadilla	Brama	Baieta
Pijota	Joputa	Baila

Tabla 2 Nombres comunes según especie³

Dado este hecho se buscó en fuentes más formales, concretamente BEDCA [18], la cual contiene una base de datos de composición de alimentos. El problema que se encontró de esta fuente fue que no solo se especificaban alimentos, sino que también incluía platos ya preparados, además de un nivel de detalle que no se correspondía a lo necesitado y una cantidad de ingredientes escasa. Así que se decidió por descartarla.

La solución definitiva se encontró cuando se consiguió una base de datos de pesticidas. Ésta contenía una lista de la mayoría de productos que se pueden encontrar en el mundo, clasificados en grupos de alimentos que tuvieran alguna relación, ya sea por pertenencia al mismo grupo científico o por relación con los pesticidas que podían aplicarse.

Finalmente se transformó la categorización de un dominio científico a un dominio que se pudiera aplicar en el mundo

de la restauración. Para ello se utilizaron diferentes libros especializados en restauración [19] [20] [21].

6.2 FoodMine

Debido a que ningún miembro del equipo es experto en restauración, nos encontramos con la necesidad de validar si la jerarquía de categorías resultante era correcta y lo suficientemente óptima, es decir, que cumpliera con lo que se buscaba: ser capaz de clasificar cualquier ingrediente que pudiera aparecer en cualquier receta.

Con lo cual se decidió diseñar e implementar FoodMine, una herramienta para que diferentes expertos pudieran interactuar directamente con la base de conocimiento. Además de la validación, mediante esta herramienta, los expertos podían traducir y cambiar las traducciones existentes de las diferentes categorías e ingredientes y añadir comentarios a un ingrediente, para justificar su existencia o los cambios realizados sobre éste.

El resultado esperado de la intervención de los expertos, era el de obtener diferentes versiones de la base de conocimiento. A partir de entonces se compararían las diferentes versiones obtenidas, y se decidiría junto con estos expertos cuál de las versiones se adaptaría mejor a las necesidades del proyecto.

El motivo principal por el cual se decidió por implementar una solución tan compleja fue por el hecho de no saber el nivel de conocimiento que los diferentes expertos podían tener en la utilización de las nuevas tecnologías y herramientas ofimáticas. Por lo tanto, se descartó la idea de utilizar hojas de cálculo o similares, que habrían dificultado enormemente el proceso de validación, traducción y categorización. Además, otro beneficio directo es el formato de los datos, ya que, al estar almacenados en una base de datos, ser modificados directamente sobre ésta y pudiéndose obtener de un modo muy rápido y fácil (mediante una API), posteriormente transferirlos a la ontología fue una tarea trivial.

6.2.1 Requisitos

Después de analizar las necesidades y las diferentes funciones que debía cubrir la aplicación se llegó a los requisitos mostrados en la Tabla 3.

ID	Definición
RF1	El sistema tiene que proporcionar un mecanismo de autenticación de usuarios.
RF2	El sistema tiene que mostrar la jerarquía de categorías.
RF3	El sistema tiene que proporcionar al usuario un mecanismo para modificar la jerarquía de categorías.
RF4	El sistema tiene que proporcionar al usuario un mecanismo para crear y eliminar categorías.
RF5	El sistema tiene que proporcionar al usuario un mecanismo para añadir y modificar las traducciones de las categorías.
RF6	El sistema tiene que mostrar los ingredientes con sus correspondientes categorías.

³ Fish Base. <http://www.fishbase.org/>.

RF7	El sistema tiene que proporcionar al usuario un mecanismo para crear y eliminar ingredientes.
RF8	El sistema tiene que proporcionar al usuario un mecanismo para añadir notas sobre los ingredientes.
RF9	El sistema tiene que proporcionar al usuario un mecanismo para añadir y modificar las traducciones de los ingredientes.
RF10	El sistema tiene que proporcionar al usuario un mecanismo para cambiar la categoría de un ingrediente.
RF11	El sistema tiene que proporcionar al usuario administrador un mecanismo para añadir nuevos usuarios.
RF12	El sistema tiene que proporcionar al usuario un mecanismo para visualizar las categorías y los ingredientes de otro usuario.

Tabla 3 Requisitos FoodMine

6.2.2 Use cases

Una vez definidos los requisitos funcionales de la aplicación, se definieron los diagramas de casos de uso.

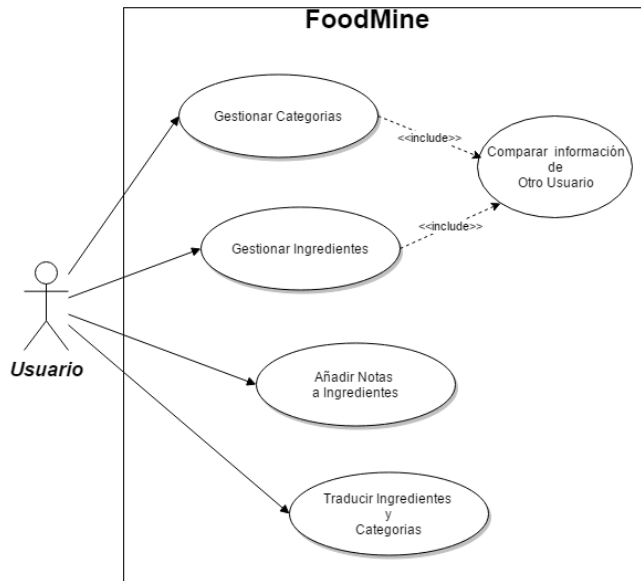


Imagen 2 FoodMine Use case

Principalmente, el usuario podrá gestionar las categorías e ingredientes, es decir, cambiarles el nombre, crear nuevos elementos y eliminarlos. En el caso de las categorías el usuario podrá cambiar la jerarquía y en el caso de los ingredientes, modificar a que categoría pertenecen. Además de añadir notas y traducir las diferentes categorías e ingredientes.

6.2.3 Prototipado

Después de definir los requisitos de la aplicación, se hicieron los prototipos de cómo debía ser visualmente. Estos prototipos principalmente sirvieron para tener una idea aproximada de cómo tenía que ser la aplicación una vez implementada.

La vista de categorías principalmente debía permitir al

usuario cambiar la jerarquía de categorías, traducirlas, eliminarlas y crear de nuevas. Éstas, se muestran en forma de árbol.

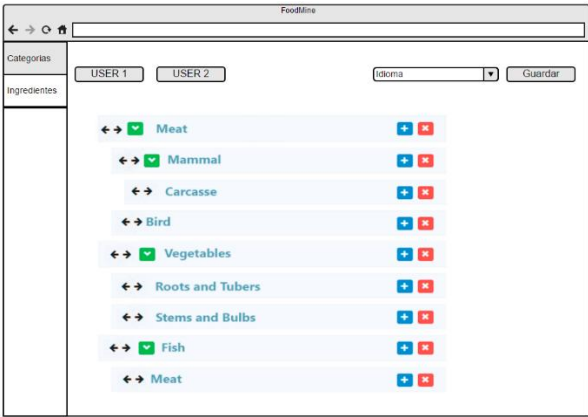


Imagen 3 Prototipo de la vista de Categorías

En la vista de ingredientes, cada ingrediente aparece en una lista con la subcategoría a la que pertenece. Además, se muestran las categorías que están por encima de la categoría del ingrediente.

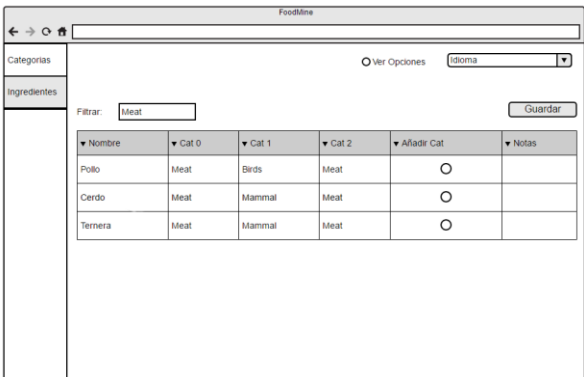


Imagen 4 Prototipo de la vista de ingredientes

Al pulsar sobre un ingrediente y teniendo marcado “Ver Opciones”, se visualizan las diferentes traducciones para ese determinado ingrediente y las notas introducidas. Además, pulsando el botón “USER 2” se visualizarían que datos a introducido ese usuario para el ingrediente en cuestión.

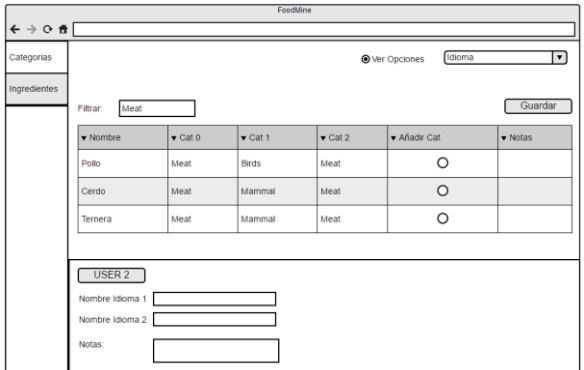


Imagen 5 Prototipo de la vista de Ingredientes con "Ver Opciones" marcado

6.2.4 Arquitectura

Una vez analizada la parte más funcional de la aplicación se diseñó la arquitectura.

Por un lado, está el backend el cual consiste en una API REST que proporciona al frontend todos los datos necesarios, además de proporcionar un mecanismo de autenticación de usuarios.

Esta autenticación se basa en el protocolo OAuth2 además de incorporar un control de acceso a los recursos (control de lectura/escritura y de propiedad).

El frontend por otro lado es una aplicación independiente, encargada de mostrar los datos proporcionados por la API y de proporcionar la interacción necesaria al usuario.

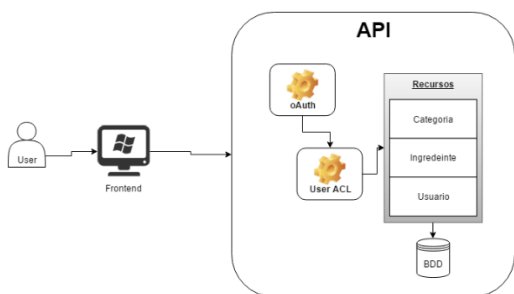


Imagen 6 Arquitectura FoodMine

6.2.5 Funcionamiento del sistema

Toda interacción entre el frontend y el backend se realiza de la misma manera.

En primer lugar, el frontend tiene que hacer una petición para obtener acceso a los diferentes recursos. Esta petición incluye un usuario, contraseña y los datos de autenticación de OAuth. Después de validar éstos, se devuelve una AccessToken que se tendrá que incluir en las siguientes peticiones.

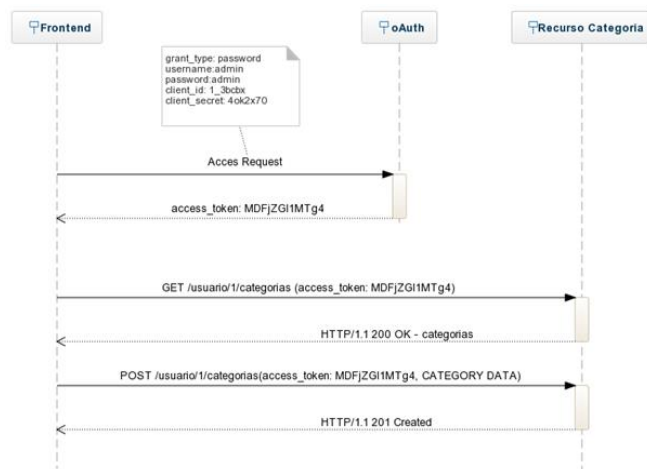


Imagen 7 Petición de un recurso

6.2.6 Base de datos del Backend y de clases

Debido a que se usó Symfony para el desarrollo del backend y éste usa un ORM para persistir los datos, el diagrama de clases y el de base de datos son iguales. Éste se puede ver en la sección A5 FoodMine Database.

6.2.7 Implementación Backend

Para la implementación de la API se usó el lenguaje PHP y el framework Symfony3.

Además, se añadieron los siguientes bundles de Symfony:

- FOSOAuthServerBundle: proporciona autenticación OAuth para toda la aplicación
- FOSUserBundle: proporciona mecanismos de control y gestión de usuarios (ACL)
- FOSRestBundle: proporciona varios mecanismos para ayudar en la construcción de aplicaciones REST.
- NelmioApiDocBundle: permite generar documentación para la API

Para la implementación de cada uno de los recursos accesibles por el frontend se siguió este workflow:

1. Creación de la entidad: clase que representa nuestro recurso.
2. Creación de las rutas: mediante las rutas estamos indicando que recursos estamos definiendo
3. Creación del controlador del recurso: se implementa la lógica para ese recurso, concretamente los métodos GET, POST, PUT y DELETE
4. Creación del formulario para la validación.
5. Creación de los test funcionales.

6.2.8 Implementación del Frontend

Para la implementación del Frontend se usó íntegramente AngularJS, un framework JavaScript. Adicionalmente se usó Yeoman, un conjunto de herramientas que facilitan el desarrollo de aplicaciones frontend.

A grandes rasgos Yeoman se encarga de crear el esqueleto de la aplicación y de crear un pequeño servidor web en el directorio de trabajo para poder probar la aplicación durante el desarrollo.

En lugar de implementar las funcionalidades del árbol de categorías, se usó una librería llamada Angular UI Tree con el objetivo de perder el menor tiempo posible implementando esta funcionalidad y probando que ésta funciona correctamente.

Con el mismo objetivo, se ha usado Materialize, un framework CSS basado en Material Design usado para darle estilo a toda la aplicación.

6.3 Ontología

A partir de los datos de FoodMine, es decir, de las categorías y de los ingredientes, se creó un script que obtiene los datos a partir de la API de FoodMine y los inserta en la ontología teniendo en cuenta los siguientes puntos:

- Cada categoría dentro de FoodMine, es una clase dentro de la ontología.
- La jerarquía definida en FoodMine es la misma que en la ontología, es decir, que se mantienen las relaciones de pertenencia entre las diferentes categorías.
- Debido a la asunción de mundo abierto (4.2.3 Open World Assumption), dos elementos por llamarse de la misma manera no tienen por qué referirse a la misma cosa. Además, el nombre con el cual se guarda un elemento, no tiene la necesidad de ser igual a como lo llamamos en lenguaje natural, ya que, un ingrediente

puede expresarse de diferentes maneras. Con lo cual, para evitar posibles problemas se ha definido una convención de nombres:

- Clases: name_c[0-9].
- Individual: name_i[0-9]

Siendo “[0-9]” un numero aleatorio de N dígitos.

- Por cada elemento e idioma, se ha creado una etiqueta con la forma en como nos referimos a éste en lenguaje natural.
- Por cada elemento se ha definido que es diferente a otros elementos del mismo tipo y nivel dentro de la jerarquía. Esto es de vital importancia, ya que al trabajar “en el mundo abierto” se ha de especificar que son diferentes, de lo contrario las inferencias podrían dar resultados inesperados.

En la Imagen 8 Ejemplo de Categoría en Protégé, se puede ver como se representaría una categoría con sus respectivos ingredientes dentro de la ontología.

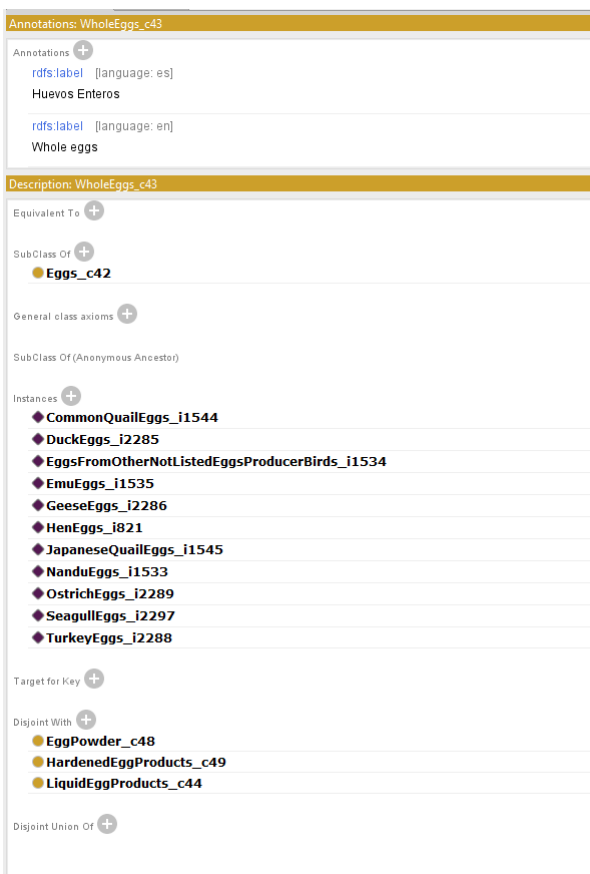


Imagen 8 Ejemplo de Categoría en Protégé

El script encargado de transferir los datos de FoodMine a la ontología se implementó utilizando Java, concretamente utilizando una librería llamada Apache Jena, la cual es un framework para la creación de aplicaciones semánticas. Más concretamente se ha usado el componente Ontology API⁴.

El workflow del script se puede ver en A3 Creación Ontología Workflow.

7 DEMO: CREACIÓN DE RECETAS

Una vez implementada la ontología se utilizó para crear diferentes axiomas y reglas para inferir conocimiento nuevo. Además, para sacarle partido a estos axiomas, se implementó una aplicación que los utiliza y les da sentido. La aplicación en cuestión es un módulo que en un futuro podría incorporarse a la *Suite* de aplicaciones para gestionar un restaurante. Concretamente, ésta se usaría para facilitar la creación de platos pertenecientes a los menús, pudiendo saber que ingredientes son necesarios, además de informar que alérgenos están presentes en cada uno de estos platos.

Toda esta información se guardaría en la ontología, siendo útil para otras aplicaciones y módulos que necesitasen de estos datos. Dentro de este contexto, la ontología formaría parte del núcleo del conjunto de estas aplicaciones, es decir, que todas ellas compartirían la misma ontología.

Volviendo al módulo en cuestión, el usuario típico de éste sería el cocinero, que lo usaría para crear o eliminar las diferentes recetas y añadir los ingredientes que pudieran faltar en el sistema. Además, éste al ir generando las diferentes recetas, podría visualizar que alérgenos están presentes en cada una de ellas.

Por lo tanto, el diagrama de casos de uso correspondiente a esta aplicación podría ser el mostrado en Imagen 9 Use case Creación recetas.

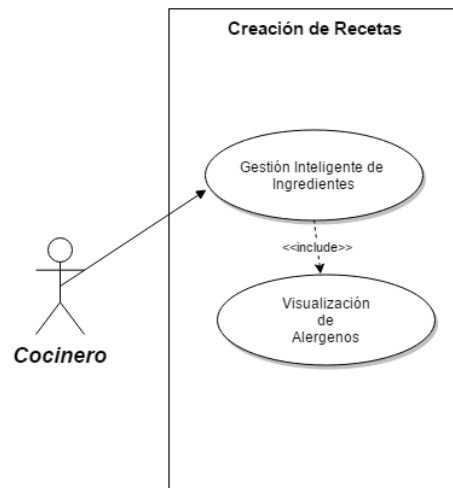


Imagen 9 Use case Creación recetas

7.1 Composición de recetas

Las funcionalidades principales que se implementaron en esta primera parte son:

- Composición de platos: en la creación de platos, el usuario puede seleccionar como ingrediente un plato ya creado, sin la necesidad de volver a introducir gran parte de los ingredientes.
- Gestionar los alérgenos: en la creación de un plato, la aplicación no necesita informar de que el plato que se va a guardar tiene X alérgenos, ya que la propia ontología será capaz de saberlo inmediatamente.

⁴ Ontology API. <https://jena.apache.org/documentation/ontology/>

Para que la ontología sea capaz de hacer la composición de platos lo primero que se hizo fue permitir la pertenencia de un ingrediente a un plato. Para ello se creó un Object Property llamado “hasIngredient”. Así podemos indicar que cierto plato X tiene un ingrediente Y.

Para crear platos a partir de otros, y que los ingredientes se muestren correctamente, se pusieron estos Object Properties como transitivos. Por ejemplo:

*“Pan hasIngredient Harina,
PanConTomate hasIngredient Pan”.*

Si ponemos hasIngredient como transitiva, la ontología inferirá:

“PanConTomate hasIngredient Harina”

Para los alérgenos, fueron necesarios diferentes elementos. El primero fue añadir a la ontología los diferentes individuals correspondientes a los diferentes alérgenos. En segundo lugar, se creó un Object Property para indicar que cierto ingrediente/clase tiene un determinado alérgeno. Por ejemplo:

“Anchovies hasAlergen Fish”.

Por último, se creó un axioma para que la asignación de nuevos alérgenos se hiciera de manera automática. Este axioma, básicamente dice que, dado un plato formado por un ingrediente o por otro plato con un alérgeno X, este también tendrá ese alérgeno X.

Para esto, se usó una única regla SWRL [16], tal y como se muestra en la Imagen 10 SWRL para la inferencia de alérgenos.



Imagen 10 SWRL para la inferencia de alérgenos

7.2 Sugerencia de Ingredientes

En un desarrollo posterior, a la aplicación se le añadieron dos nuevas funcionalidades:

- A medida que el usuario va introduciendo ingredientes, el sistema a partir de las recetas que ya ha aprendido, propone ingredientes que en otras recetas se han combinado con los ingredientes que el usuario ha ido escogiendo.
- Durante la creación de una receta se puede encontrar posibles substitutos a un ingrediente.

Para el desarrollo de estas nuevas funcionalidades, prácticamente no se tuvo que hacer nada en la ontología, únicamente se añadió un nuevo Object Property: “isIngredientOf”, haciéndolo transitivo e inverso al definido anteriormente “hasIngredient”. Volviendo al ejemplo anterior, la ontología inferiría:

“Harina isIngredientOf Pan”

Además, para estas nuevas funcionalidades se usaron diferentes consultas SPARQL [22] [23] [24].

Para sugerir los ingredientes que en alguna receta se han combinado con los ingredientes que el usuario ha elegido, se usó la consulta de la Imagen 11, donde se muestra como se buscan los ingredientes que se han combinado con “Buns” y “CalfFreshMeat”.

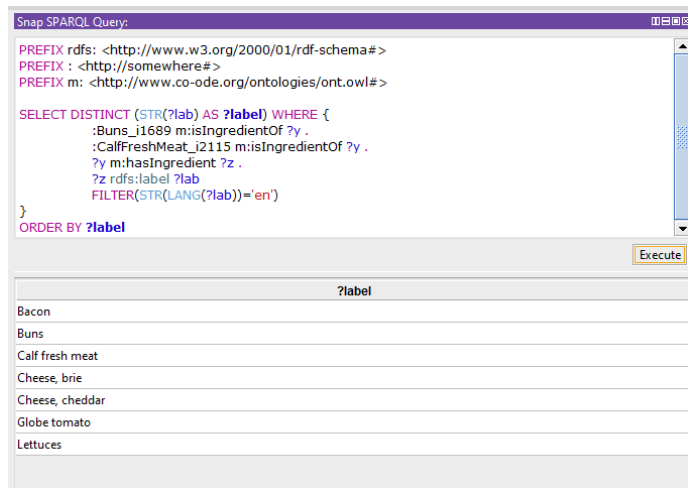


Imagen 11 Consulta SPARQL para obtener las posibles combinaciones

Como se ha comentado anteriormente, al transformar la base de datos de pesticidas a un dominio de restauración, todos los ingredientes se clasificaron según propiedades y tipos. Así, dado un ingrediente dentro de una categoría, es más que probable que el resto de ingredientes dentro de esa misma categoría sean muy parecidos. Es el caso del queso Brie: dentro de su categoría se encuentran otros quesos como Camembert, Chaource o Coulommiers. Éstos son muy parecidos al Brie en la textura y en el sabor, aunque la decisión final de escoger uno u otro dependerá del usuario. Dado este caso, listar todos los quesos parecidos al Brie, es tan sencillo como se muestra en la Imagen 12.

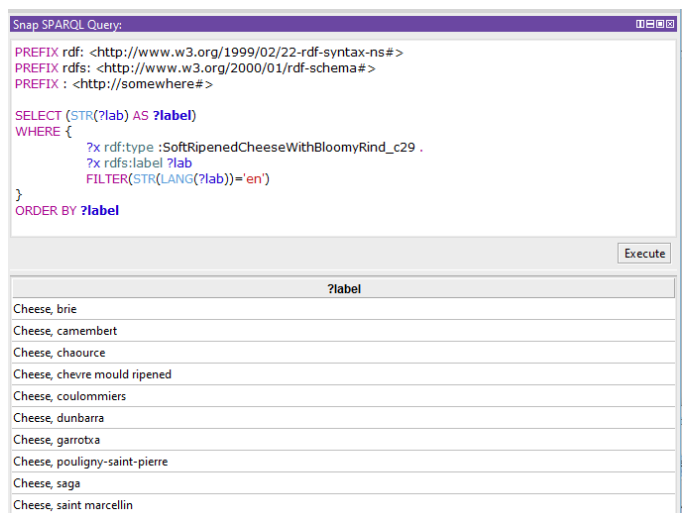


Imagen 12 Consulta SPARQL para buscar substitutos

7.3 Resultados en la Ontología

Si ejecutásemos la aplicación y creásemos por ejemplo una pizza margarita, en la ontología se vería reflejado tal y como se muestra en la Imagen 13.

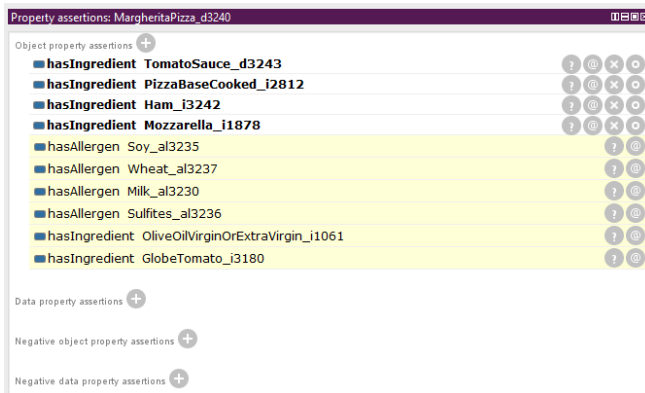


Imagen 13 Resultados en la Ontología

8 CONCLUSIONES Y RESULTADOS

Llegados al último tramo del desarrollo del trabajo, se podría decir que se han cumplido los objetivos casi en su totalidad.

En primer lugar, se tenía que obtener la base de conocimiento para la ontología. Como se ha visto, ésta acabó siendo una base de datos de pesticidas, lo que nos facilitó el trabajo a realizar, teniendo que reestructurar ésta en un dominio de restauración.

En segundo lugar, se debería haber validado la base de conocimiento. La herramienta para validar la base de conocimiento si se desarrolló (FoodMine), pero la validación por parte de los expertos no se realizó por falta de tiempo. A pesar de este hecho, se continuó con el desarrollo del proyecto.

En tercer lugar, se creó la primera versión de la ontología, ya que el script para crearla no dependía del estado de la base de conocimiento.

Finalmente, se crearon diferentes axiomas, que se utilizaron para crear la aplicación de creación de recetas, la cual como se ha visto, utiliza estos axiomas para hacer diferentes inferencias.

Referente a la planificación del proyecto, debido a la metodología elegida, se ha ido ajustando a medida que se ha ido avanzando, siempre teniendo en cuenta las milestones definidas.

El hecho de llevar a cabo este trabajo dentro de la empresa Delectatech ha supuesto ventajas como el acceso a los diferentes expertos, llevar una planificación más estricta y tener acceso a las necesidades del proyecto más directamente. Por otro lado, ha hecho que algunos procesos tardaran más de la cuenta, como por ejemplo la validación de la base de conocimiento, ya que no se ha llevado a cabo debido a que no ha dado tiempo a que las diferentes partes se pongan de acuerdo.

Personalmente este trabajo ha sido un reto, ya que nunca había realizado nada relacionado con aprendizaje ni procesamiento de lenguaje natural de manera tan formal. Además, he aprendido unos conceptos y unas técnicas que muy posiblemente pueda aplicar en muchos otros campos.

9 AGRADECIMIENTOS

Me gustaría agradecer en primer lugar al tutor de este trabajo, Dimosthenis, todo su interés y preocupación por que el proyecto siguiera adelante.

En segundo lugar, quiero agradecer a todo el equipo de Delectatech, primero la oportunidad de realizar este proyecto y segundo toda su colaboración y paciencia. Por último y no menos importante, a todas las personas que han aportado algo a este proyecto.

10 BIBLIOGRAFÍA

- [1] T. R. Gruber, «A translation approach to portable ontology specifications,» *Knowledge Acquisition*, vol. 5, pp. 199-220, 1993.
- [2] S. Staab y R. Studer, *The Handbook on Ontologies in Information Systems*, Berlin: Springer, 2003, pp. 51-66.
- [3] D. Bruce Lenat y R. V. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*, Addison-Wesley, Boston, 1990.
- [4] M. Uschold y M. King, «Towards a Methodology for Building Ontologies,» de *IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, 1995.
- [5] M. Grüninger y M. S. Fox, «Methodology for the design and evaluation of ontologies,» de *Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, 1995.
- [6] A. Barnaras, I. Laresgoiti y J. Corera, «Building and reusing ontologies for electrical network,» de *Proc. European Conference on Artificial Intelligence (ECAI-96)*, Budapest, Hungary, 1996, pp. 298-302.
- [7] A. Gomez-Perez, M. Fernandez-Lopez y A. de Vicente, «Towards a Method to Conceptualize Domain Ontologies,» de *ECAI96 Workshop on Ontological Engineering*, Budapest, 1996, pp. 41-51.
- [8] B. Swartout, B. Ramesh, K. Knight, T. Russ, P. Ramesh y K. Swartout, «Toward Distributed Use of Large-Scale Ontologies,» de *AAAI Symposium on Ontological Engineering*, Stanford, 1997.
- [9] S. Staab, P. Schnurr H., R. Studer y Y. Sure, «Knowledge processes and ontologies,» de *IEEE Intelligent Systems 16*, 2001, pp. 26-34.
- [10] O. Corcho, M. Fernandez-lopez y A. Gomez-Perez, «Methodologies, tools and languages for building ontologies. Where is their meeting point?,» de *Data & Knowledge Engineering 46*, 2003, pp. 41-64.

- [11] N. F. Noy y D. L. McGuinness, «Ontology Development 101: A Guide to Creating Your First Ontology,» Stanford University, Stanford, CA, 94305, 2001.
- [12] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider y S. Rudolph, «OWL 2 Web Ontology Language,» 11 Diciembre 2012. [En línea]. Available: <https://www.w3.org/TR/owl2-primer/>. [Último acceso: 2 Mayo 2016].
- [13] B. Cuenca Graua, I. Horrocks, B. Motika, B. Parsia, P. Patel-Schneider y U. Sattler, «OWL 2: The next step for OWL,» *Web Semantics: Science, Services and Agents*, pp. 309-322, 2008.
- [14] S. Staab y R. Studer, «Web Ontology Language,» de *Handbook on Ontologies*, 2009, pp. 67-92.
- [15] R. Stevens, «How does a reasoner work?,» 12 Agosto 2014. [En línea]. Available: <http://ontogenesis.knowledgeblog.org/1486>. [Último acceso: 1 Mayo 2016].
- [16] Stanford University, «SWRLLanguageFAQ,» 23 Julio 2014. [En línea]. Available: <http://protege.cim3.net/cgi-bin/wiki.pl?SWRLLanguageFAQ>. [Último acceso: 10 Mayo 2016].
- [17] R. Gulizzia, «IBM's Watson: Designed to Learn Like a Human: Institute for Creation Research,» 19 Marzo 2014. [En línea]. Available: <http://www.icr.org/article/8030/>. [Último acceso: 2016 Mayo 26].
- [18] Ministerio de Ciencia e innovación, «Base de Datos Española de Composición de Alimentos,» [En línea]. Available: <http://www.bedca.net/>. [Último acceso: 20 3 2016].
- [19] Larousse, Le Grand Larousse Gastronomique, Larousse, 2007.
- [20] H. McGee, On Food And Cooking The science and Lore of the Kitchen, New York: Scribner, 2004.
- [21] R. Rabaso y F. Aneiros, El Práctico Resumen mundial de cocina y pastelería, Buenos Aires: Librería Universitaria.
- [22] E. Sirin y B. Parsia, «SPARQL-DL: SPARQL Query for OWL-DL,» Manchester, UK, 2007.
- [23] J. Pérez, M. Arenas y C. Gutierrez, «Semantics and complexity of SPARQL,» *ACM Transactions on Database Systems*, 2009.
- [24] S. Hawke, I. Herman, B. Parsia, A. Polleres y A. Seaborne, «SPARQL 1.1 Entailment Regimes,» W3C, 21 Marzo 2013. [En línea]. Available: <https://www.w3.org/TR/sparql11-entailment/>. [Último acceso: 12 Mayo 2016].

APÉNDICE

A1 Log planificación

- En progreso: al finalizar el sprint esa tarea no se finalizó, quedando pendiente para el siguiente sprint.

- Finalizado: al finalizar el sprint esa tarea se consideró terminada.

Semana del 8 al 14 Febrero		
Id. Tarea	Nombre	Estado
0.1	Formación sobre ontologías	En Progreso
0.3	Definición de los objetivos	Finalizado
0.4	Primera planificación	Finalizado

Semana del 15 al 21 de Febrero		
Id. Tarea	Nombre	Estado
0.1	Formación sobre ontologías	En Progreso
0.5	Elaboración del informe inicial	En Progreso

Semana del 22 al 28 Febrero		
Id. Tarea	Nombre	Estado
0.1	Formación sobre ontologías	En Progreso
0.5	Elaboración del informe inicial	En Progreso
0.2	Formación sobre OWL	En Progreso

Semana del 29 de Febrero al 6 de Marzo		
Id. Tarea	Nombre	Estado
0.1	Formación sobre ontologías	Finalizado
0.5	Elaboración del informe inicial	Finalizado
0.2	Formación sobre OWL	Finalizado

Semana del 7 de Marzo al 13 de Marzo		
Id. Tarea	Nombre	Estado
1.1	Busqueda de fuentes	En Progreso
1.2	Analisis de fuentes	En Progreso

Semana del 14 de Marzo al 20 de Marzo		
Id. Tarea	Nombre	Estado
1.1	Busqueda de fuentes	Finalizado
1.2	Analisis de fuentes	Finalizado

Semana del 21 de Marzo al 27 de Marzo		
Id. Tarea	Nombre	Estado
2.1.1	FoodMine: análisis de requisitos y casos de uso	Finalizado
2.1.2	FoodMine: Diseño de la arquitectura	Finalizado
2.1.3	FoodMine: Diagrama de Secuencia	Finalizado
2.1.4	FoodMine: Diseño BDD y Clases	Finalizado
2.2	FoodMine: Formación Symfony	En Progreso

Semana del 28 de Marzo al 3 de Abril		
Id. Tarea	Nombre	Estado
2.2	FoodMine: Formación Symfony	Finalizado
2.3.1	FoodMine: Implementación Base API	Finalizado
2.3.2.1	FoodMine: Recurso Usuario	En Progreso

Semana del 4 de Abril al 10 de Abril		
Id. Tarea	Nombre	Estado
2.3.2.1	FoodMine: Recurso Usuario	Finalizado
2.3.2.2	FoodMine: Recurso Categoria	En Progreso
0.6	Elaboración Informe Progreso I	En Progreso

Semana del 11 de Abril al 17 de Abril		
Id. Tarea	Nombre	Estado
2.3.2.2	FoodMine: Recurso Categoria	Finalizado
2.3.2.3	FoodMine: Recurso Ingrediente	Finalizado
2.4	FoodMine: Implementación e Integración Frontend-Backend	Finalizado
0.6	Elaboración Informe Progreso I	Finalizado

Semana del 18 de Abril al 24 de Abril		
Id. Tarea	Nombre	Estado
0.2	Formación sobre OWL	En Progreso
0.10	Formación Apache JENA	En Progreso

Semana del 25 de Abril al 1 de Maig		
Id. Tarea	Nombre	Estado
0.2	Formación sobre OWL	En Progreso
0.10	Formación Apache JENA	En Progreso

Semana del 2 de Maig al 8 de Maig		
Id. Tarea	Nombre	Estado
0.2	Formación sobre OWL	En Progreso
0.10	Formación Apache JENA	Finalizado
0.2	Formación sobre OWL	Finalizado

Semana del 9 de Maig al 15 de Maig		
Id. Tarea	Nombre	Estado
0.11	Formacion Protégé	Finalizado
3	Creacion de la Ontologia	Finalizado

Semana del 16 de Maig al 22 de Maig		
Id. Tarea	Nombre	Estado
0.7	Elaboración Informe Progreso II	Finalizado
4	Creación Demo 1	Finalizado

Semana del 23 de Maig al 29 de Maig		
Id. Tarea	Nombre	Estado
0.9	Elaboración articulo	En Progreso
4	Creación Demo 2	Finalizado

Semana del 30 de Maig al 5 de Juny		
Id. Tarea	Nombre	Estado
0.9	Elaboración articulo	En Progreso

Semana del 6 de Juny al 12 de Juny		
Id. Tarea	Nombre	Estado
0.9	Elaboración articulo	En Progreso

Semana del 13 de Juny al 19 de Juny		
Id. Tarea	Nombre	Estado
0.9	Elaboración articulo	En Progreso

Semana del 20 de Juny al 26 de Juny		
Id. Tarea	Nombre	Estado
0.9	Elaboración articulo	Finalizado

A2 Milestones

Milestone 1 - 17 abril 2016	
Nombre	Descripción
Informe de progreso 1 realizado	Primer informe de progreso redactado.
Fuentes de información definidas y analizadas	Las fuentes de información son conocidas y se sabe qué información ofrece cada una.
Primera Fase Implementación Food-Mine	Implementada la base y los recursos de Usuario y Categoría.

Milestone 2 - 22 mayo 2016	
Nombre	Descripción
Informe de progreso 2 realizado	Segundo informe de progreso redactado.
Implementación FoodMine Finalizada	FoodMine completamente acabada.
Herramienta implementada para la transferencia de la información a la Ontología.	La herramienta o script para transferir la información clasificada a la Ontología.

Milestone 3 - 17 junio 2016	
Nombre	Descripción
Artículo y dossier realizado	Artículo y dossier terminados.
Ontología preparada para ser validada.	La información transferida a la Ontología
validación de la Ontología	Base de conocimiento validada.
Inferencias y demostraciones	Se realizaran diferentes demostraciones e inferencias.

A3 Creación Ontología Workflow

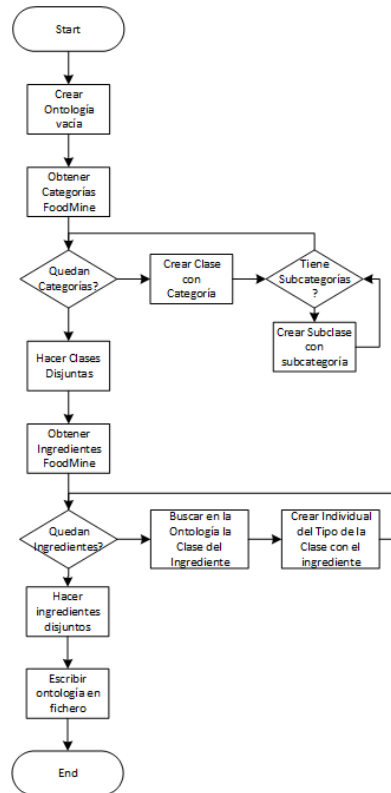


Imagen 14 Workflow Creación Ontología

A5 FoodMine Database

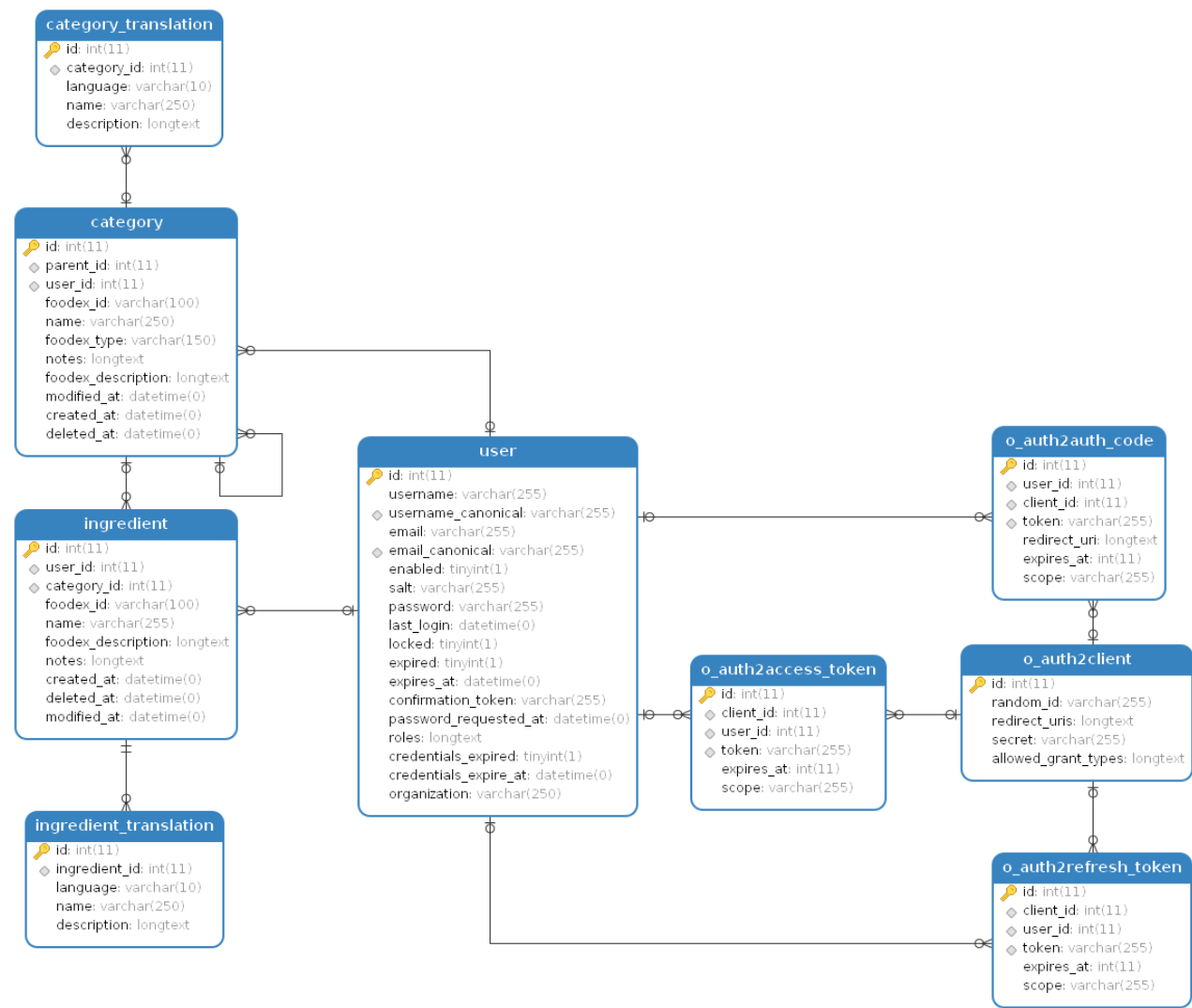


Imagen 15 FoodMine database